Clicbait Detection Using Naïve Bayes Algorithm and XAI

Haq, Md. Ramim Ul
Student
BRAC University
Dhaka, Bangladesh
ramimmd1@gmail.com

Toki, Sadikul Alim
Student
BRAC University
Dhaka, Bangladesh
sadikul.alim.toki@g.bracu.ac.bd

Haq, Quazi Shahriar
Student
BRAC University
Dhaka, Bangladesh
quazi.shahriar.haq@g.bracu.ac.bd

Rashid, Sk. Mamunur
Student
BRAC University
Dhaka, Bangladesh
sk.mamunur.rashid@g.bracu.ac.bd

***Abstract*— This paper is based on an AI algorithm named naïve bayes to filter out clickbait online links for the users to internet without falling into misleading or sensational news. Clickbait is a link which attracts and encourages the users to click on that particular link which leads to a particular website to spread false or misleading information online. We in this paper try to filter out the clickbait using naïve bayes algorithm with a manually entered dataset.**

Keywords*—*clickbait, non-clickbait, AI, algorithm, detection, naïve bayes, algorithm, XAI etc.

I. Introduction

Clickbait is a link containing short messages leads users to link on the content and direct to another website luring them into sensational and misleading information headings like:

i. Things you need to know

ii. You will never trust again

iii. This trick may hack your lifestyle

It’s kind of a tabloid journalism [1] refers to scandal, sensational news to attract most users to direct them to websites often containing viruses. Those websites earn revenue based on page views [2]. Clickbait links are to be appear on some various webpages with short size ads and by clicking it the user is directed to the desired website [3]. In this paper we are using naïve bayes AI algorithm model to filter out the clickbait news and links so that the spread of fake news stops and don’t direct the users to spamming sites. The clickbait is referred as 1 and non-clickbait are referred as 0. Our dataset consists of 32000 of data where clickbait and non-clickbait percentage is 50-50. Our main goal is to classify the clickbait and non-clickbait sites using naïve bayes algorithm and explain the AI model using an explainable AI with our manually dataset. We implemented lime XAI here in the model to describe the code.

II. Clickbait and Non-Clickbait

In here we have gathered the dataset from Kaggle [4] which is based on various sources about the clickbait and non-clickbait Headlines (52000 data) according to Buzzfeed, NY Times, Upworthy, The Guardian, The Hindu etc. (shown in the chart).

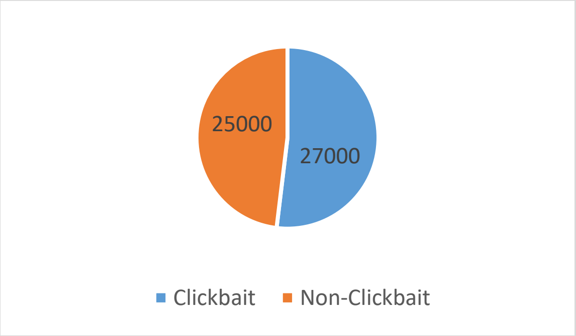


Figure 1: No. of Headlines

We will be applying our naïve bayes algorithm in this dataset in order to classify the data from clickbait to non-clickbait with number 0 and 1. As well as we will be using Lime XAI. A closer probe of the headlines in clickbait gives us an understanding that same words or data occur more frequently than the non-clickbait headlines.

III. Naïve Bayes classifier

Naïve Bayes classifier (NBC) is a simplified Bayesian probability model where the probability of one feature/variable is not affected by any other feature/variable. Let us assume that $X= (x_1, x_2, x_3, \ldots\ldots\ldots\ldots, x_n)$ represents n features/variables of a problem instance for a K class classification problem. Naïve Bayes assigns X a probability $p(C_k|x_1, x_2, x_3, \ldots\ldots\ldots\ldots\ldots, x_n)$, where $k \in\{1,2,\ldots\ldots\ldots\ldots k\}$.

NBC uses maximum a-posterior decision rule to assign a class $\hat{y}$ to an instance X with n features/variables as following:

$$\hat{y}= \underset{k\in\{1,\,2,\ldots\ldots K\}}{\arg\max}\; p\,(C_k)\prod_{i=1}^{n} p(x_i|C_k)$$

IV. Methodology

This section has discussed in details about the complete methodology of the model. Google's CoLab environment has been used to engineer the complete model. Below the sequential method has been discussed:

i) Importing tools and loading the dataset

In python requires importing tools to the model before executing. The main libraries we used are Numpy, Pandas, NLTK (Natural language toolkit) and Scikit-learn. It has also been mentioned earlier that the dataset used has been taken from Kaggle. Therefor, it was required to download the dataset in Google Drive as a compressed file. Then we have unzipped the file with *!unzip* to obtain our *.csv* file. The file *clickbait_data.csv* has been then loaded in the model.

ii) Splitting and Analyzing Train and Test

The dataset is splitted into 75% Training and 25% Testing. This enables us to verify our model and provide enough information to get trained. After analyzing it is seen that 24000 data are being used for Training and 8000 data will be used for testing the model.

iii) Pre-processing of data in several steps

A machine does not understand sentences. Hence, it is required to convert the words and sentences into quantitative units. These quantities are understandable by the machine model. However, there are several steps to go through before converting words and sentences to quantities. Here the steps have been given:

a) Tokenization of Data:

The data is tokenized i.e. split into tokens which are the smallest or minimal meaningful units. The data is split into words.

b) Converting to lowercase:

The data is converted into lowercase to avoid ambiguity between same words in different cases like 'NLP', 'nlp' or 'Nlp'.

c) Removing punctuation:

The punctuations are removed to increase the efficiency of the model. They are irrelevant because they provide no added information.

d) Removing Numbers, stop-words and extra Spaces

e) Lemmatization:

Lemmatization in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. It involves the morphological analysis of words.

In lemmatization we find the root word or base form of the word rather than just clipping some characters from the end e.g. is, are, am are all converted to its base form be in Lemmatization

Here lemmatization is done using NLTK library.

f) Conversion of texts into features:

We have TF-IDF (Term frequency-Inverse Data Frequency) from *sklearn library* to convert the texts into features. These features would be used to classify the sentences into Clickbaits and non-Clickbaits. In our model we have used both bi-gram and tri-gram extraction methods to compare the accuracy

g) Training with *Multinomial Naive Bayes classification algorithm:*

The classification comprises two phases: the learning phase and the evaluation phase. A specified dataset is used to train the classifier's model during the learning stage, and performance is checked during the evaluation step. Performance is assessed using a variety of criteria, including accuracy, error, precision, and recall rate. It has been discussed earlier that the text data has been splitted into a training set and a testing set using scikit-learn. In Fig-2, the flow diagram shows the process.

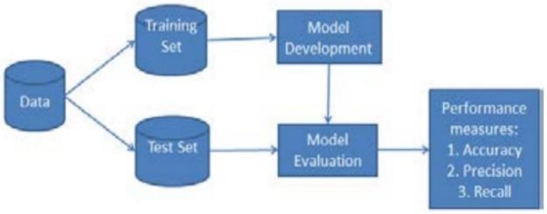


Figure 2: Multinomial Naive Bayes Classification Model

V. Result Analysis

i) Confusion Matrix and the results for Tri-gram extraction has been given below:

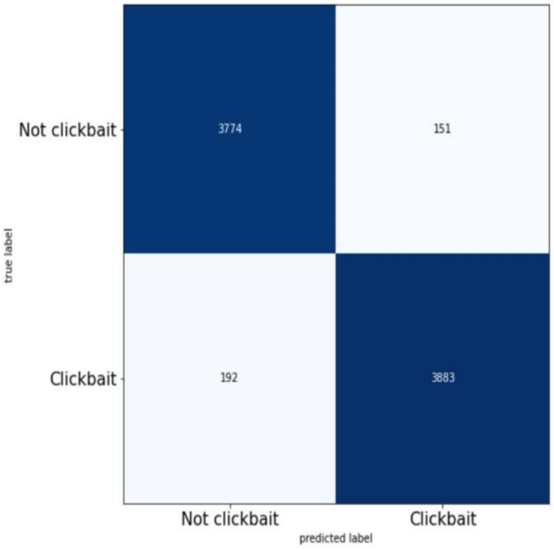
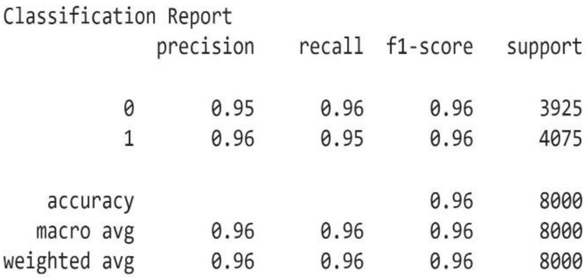


Figure 4 Confusion Matrix for Tri-gram extraction



```
Classification Report
              precision    recall  f1-score   support

           0       0.95      0.96      0.96      3925
           1       0.96      0.95      0.96      4075

    accuracy                           0.96      8000
   macro avg       0.96      0.96      0.96      8000
weighted avg       0.96      0.96      0.96      8000
```

Figure 4 Confusion Matrix for Tri-gram extraction

ii) Confusion Matrix and the results for Bi-gram extraction has been given below:

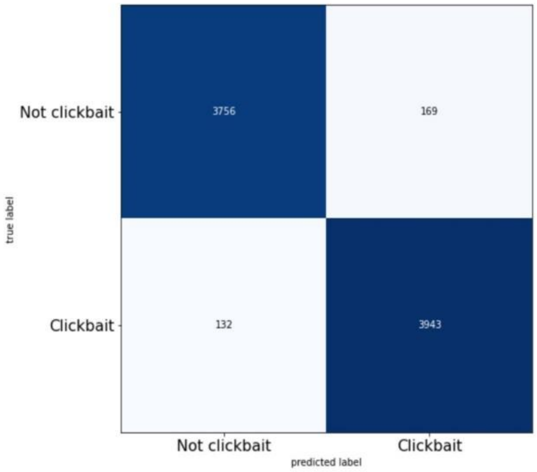
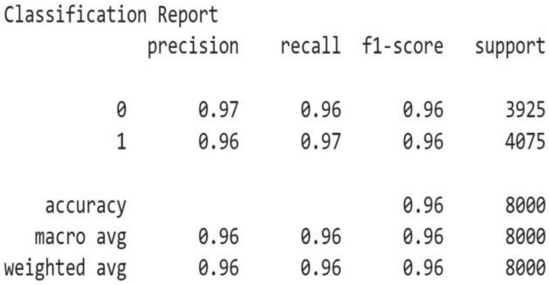


Figure 3 The results for Bi-gram extraction



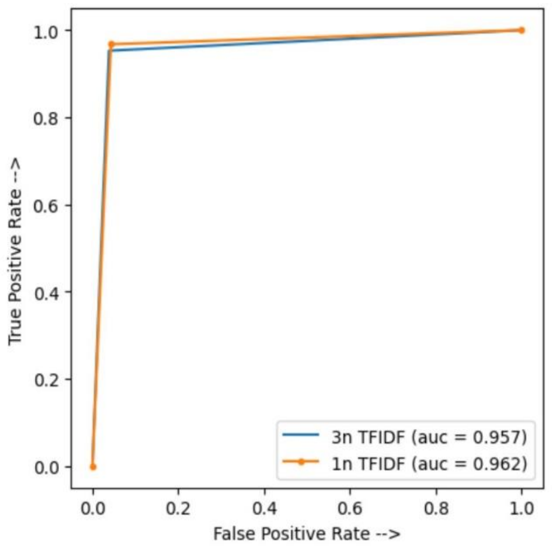
```
Classification Report
              precision    recall  f1-score   support

           0       0.97      0.96      0.96      3925
           1       0.96      0.97      0.96      4075

    accuracy                           0.96      8000
   macro avg       0.96      0.96      0.96      8000
weighted avg       0.96      0.96      0.96      8000
```

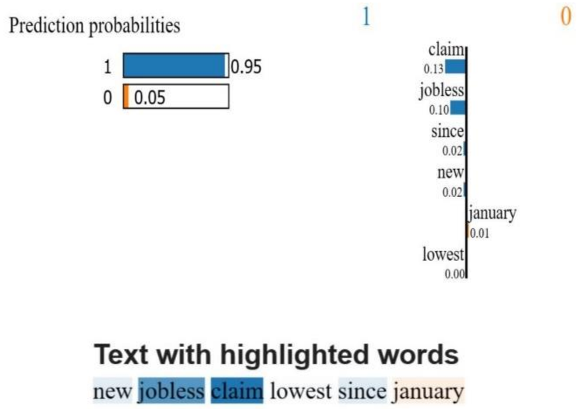
Figure 5 The results for Bi-gram extraction

From the above Classification reports and Confusion Matrices it is easy to understand that taking features either in Bi-gram or Tri-gram in TF-IDF does not change much in the output results. We have tried to make a comparison of this output using a RoC-AuC Curve.



VI. Explaining with LIME XAI

As we are trying to show our model's result through Explainable Artificial Intelligence, in order to make it human interpretable, we are using LIME (Local Interpretable Model-agnostic Explanations).



As we can see the result of LIME in the figure of our trained model, it shows which word contributes how much to the sentence being clickbait or not. For the sentence "new jobless claim lowest since January", the words "claim" and "jobless" are most responsible for being the sentence a clickbait. The result shows that the sentence is 95% clickbait according to our trained model.

VII. Conclusion

In this paper, we tried to train a model based on Multinominal Naive Bayes Classifier to detect Clickbait in sentences. Our model gave a wonderful accuracy of 96% in Bi-gram TF-IDF feature conversion and 95% accuracy in Tri-gram TF-IDF feature conversion. We have also observed that with increase in n-gram TF-IDF the accuracy for classification decreases. Along with that, we also implemented LIME as a part of our Explainable A.I. results. The MNB algorithm is a nearly modern text categorization system that is quick, simple, and easy to use. The proposed approach and algorithm provide a wide range of text categorization options. Additionally, our classifier can be modified in a few ways to improve accuracy. In order to maximize accuracy, it would be done in the future and incorporate the use of artificial intelligence.

VIII. References

[1] Chen, Y., Conroy, N. J., & Rubin, V. L. (2015, November). Misleading online content: recognizing clickbait as" false news". In Proceedings of the 2015 ACM on workshop on multimodal deception detection (pp. 15-19).

[2] Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016, March). Clickbait detection. In European conference on information retrieval (pp. 810Fr-817). Springer, Cham.

[3] Chakraborty, A., Paranjape, B., Kakarla, S., & Ganguly, N. (2016, August). Stop clickbait: Detecting and preventing clickbaits in online news media. In 2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM) (pp. 9-16). IEEE.

[4] kaggle.com/datasets/amananandrai/clickbait-dataset

[5] Multinomial Naive Bayes Classification Model for Sentiment, Analysis Muhammad Abbas , Kamran Ali Memon, Abdul Aleem Jamali, Saleemullah Memon, Anees Ahmed, IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.3, March 2019